Professional Elective 2

EIT Elective 2 – 1

WTh 8:00 am – 9:30 am

Group no. 7

Chapter 4: Use Case Analysis

Alvarez, Ansherina

De Villar, Vince Anfernee

Dela Pena, Angeline Kassandra

Fidelino, Julian Marcus

Maramara, Sophia

Molina, Atasha Rich

Torralba, Alexander James

Prof KT V. Fortuny

Professor

**Introduction**

Use cases assist us in comprehending and clarifying the users' required interactions with the system and can reveal most, if not all, of the new system's functional requirements. As a result, use cases are extensively used in the analysis phase when working with users in interviews or workshop settings to discover user and functional requirements.

**Use Cases**

* Use cases originated as a part of the object-oriented development world but have been accepted as a useful tool regardless of the development methodology in use.
* Use cases describes what the system will do from the user's perspective.
* Use cases depicts a set of activities performed to produce some output result.
* Creation of use cases is often done as a part of interview sessions with users and as a part of JAD (Joint Application Development) sessions.

**Characteristics of Use Cases**

* Organize functional requirements.
* Models the goals of user-system interactions.
* Records paths called scenarios from trigger events to goals.
* Describe one main flow of events and various alternate flows.
* Multi-level.

**Elements of a Use Case**

* Basic Information
  + Each use case has a name and a number assigned to it. The name should be as brief as possible the number is merely a numerical sequence that is used to identify each use case.
    - Priority
    - Actor
    - Trigger
* Preconditions
  + Define the state the system must be in before the use case commences.
* Normal Course
  + The normal course lists the steps that are performed when everything flows smoothly in the system. This is sometimes called the “happy path”.
* Alternative Courses
  + Alternative courses are included to depict branches in logic that also will lead to a successful conclusion of the use case.

Text

Description automatically generated

* Postconditions
  + Post-conditions specify what the use case must do before it can be terminated. These activities may occasionally be part of the normal and exceptional event flows.
* Exceptions
  + Are error conditions encountered while performing use case steps.
* Summary Inputs and Outputs
  + The use case's final section summarizes the major inputs and outputs to the use case's steps. Each of the use case's major inputs and outputs, as well as their source and destination, are listed.
* Additional Use Case Issues
  + Additional sections on use case forms may be included by some organizations. It may be useful to include sections devoted to:
    - Frequency of use
    - Business Rules
    - Special Requirements
    - Assumptions
    - Notes and Issues

Text

Description automatically generated with medium confidence

**Alternative Use Case Formats**

* Request a Chemical Use Case
  + a **fully dressed use case**
  + very thorough, detailed, and highly structured
  + written as an essential use case
  + it depicts the user-system interactions as abstract, technology-independent steps

Table

Description automatically generated

* Fully Dressed Use Case
  + Is not always required but does provide value in certain circumstances. Fully dressed use cases are especially valuable when:
    - User representatives are not closely engaged with the development team throughout the project.
    - The application is complex and has a high risk associated with system failures.
    - Comprehensive test cases will be based on the user requirements.
    - Collaborating remote teams need a detailed, shared understanding of the user requirements.

**Casual Use Case Format**

Text

Description automatically generated with medium confidence

**Use Cases and the Functional Requirements**

* Functional Requirements
  + DOs and DONTs of the system
  + It is the things that are placed concisely that aims to know what the system should provide to accomplish the user's tasks.

Where can we obtain the data for the functional requirements?

ANSWER: **based on the stories or observations of the potential users of the system**

How can we get stories or observations from the potential users of the system?

ANSWER: **Use Cases**

Why are functional requirements essential?

* Stakeholders **have a single source of truth**
* High predictability = saves time and resources
* Helps developers see if the system caters all the specific functionalities

Examples:

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Use Cases and Testing**

A technique for identifying test cases that cover the entire system from start to finish on a transaction-by-transaction basis. Many organizations develop test plans early in the development process. This strategy gives the testing/quality assurance personnel an early understanding of the system underdevelopment. It also allows them to readily identify elements of the tests they will want to perform when the system is put through its paces.

**Building Use Cases**

* Identify the Major Use Cases
  + begins with the requirement definition:
    - **Process-oriented functional requirements** - things the system must do
    - **Information-oriented functional requirements** - content the system must have
  + Activities
    - Start a use case report form for each use case by filling in the name, description, and trigger.
    - If there are more than nine use cases, group them into packages.
  + Typical Questions Asked
    - Ask who, what, when, and where about the use cases (or tasks)
    - What are the major tasks that are performed?
    - What triggers this task? What tells you to perform this task?
* Identify the Major Steps for Each Use Case
  + Here are the following steps:
    - **Complete the main body of the use case form -** users and analysts work together to **describe the envisioned interactions between the user and the system** to complete the response to the event.
    - **The user-system interactions should be outlined as a series of steps in the Normal Course section of the form -** focus on what an independent observer would see the user and system do in response to the event
* Identify Elements withing Steps
  + The goal at this point is to identify the major inputs and outputs for each step.
  + Identify detailed information for the steps but provide only general categories in the summary area of the use case form.

Examples of elements within steps:

Text

Description automatically generated

Table

Description automatically generated

**Confirm the Use Case**

* Confirm that the use case is correct as written
* Review the use case with the users to make sure that each step and each **input and output are correct** and that the final result of the use case is **consistent**.
* Ask the user to **role-play, or execute the use case** by using the written steps in the use case.

The user follows the written steps like a recipe to make sure that those steps and inputs really can produce the outputs and final result defined for the use case.

Graphical user interface, text

Description automatically generated

**Revise Functional Requirements Based on Use Cases**

The functional requirements in the requirements definition may be modified to reflect this more detailed understanding and to provide insight to the development team on some “back-end” processing that will be needed that may not be obvious from the use cases alone.

Graphical user interface, text, application

Description automatically generated

**Applying the Concepts at Tune Source**

The first step in creating the use cases is to identify the major use cases according to the requirements definition, which was developed in the last chapter. Take a minute and carefully read the requirements definition. Identify the major use cases that you think need additional definition before you continue reading. The information in the functional requirements definition sometimes just flows into the use cases, but it usually requires some thought as to how to structure the use cases. After you read the requirements definition, you may be tempted to identify use cases that correspond directly to the requirement categories, such as (1) search and browse, (2) purchase, and (3) promote.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Elaborating on the Use Cases**

The goal at this point is to describe how the use case operates. The best way to begin to understand these use cases is to visualize yourself browsing a sales-oriented Web site, searching for particular items, investigating specific items further, finally making a decision to buy, and completing the purchase. The techniques of visualizing your interaction with the process and thinking about how other systems work (informal benchmarking) are important techniques that help analysts and users understand how processes work and how to write the use cases.

* Search and Browse Tunes
  + The Tune shopper is intended to look around the tunes website but not necessarily includes for the purpose of purchasing any of the tunes.

Table

Description automatically generated

* Purchase Tunes
  + The Tune Buyer actor is specified for the Purchase Tunes use case. This designation is given because the website user has indicated a desire to purchase the item(s) in the cart.

Table

Description automatically generated

* Promote Tunes
  + As for the Promote Tunes use case, the marketing team analyzes the files of recent customer purchases and additions to the customer favorites list on a regular basis.

Table

Description automatically generated